



Sección de dispositivos móviles  
abril de 2019

## Desarrollo Multiplataforma



### 1. Introducción

Debido al impresionante avance de los dispositivos móviles en la vida cotidiana de las personas, desarrollar software se ha hecho una tarea más complicada para las personas encargadas de ello debido a que ahora se debe tomar en cuenta filosofías como “mobile first”, con el fin de que a cada cliente se les pueda proporcionar los mismos servicios en sus distintos y variados dispositivos electrónicos.

### 2. ¿Qué es una aplicación multiplataforma

Empecemos por entender a que se le denomina aplicación multiplataforma (o cross-platform mobile app), una aplicación multiplataforma es aquella que puede ser ejecutada en más de una arquitectura de hardware. Básicamente existen dos tipos de aplicaciones multiplataforma:

1. Aplicación multiplataforma nativa (Native Cross-Platform Apps)
2. Aplicación multiplataforma híbrida ‘HTML5’ (Hybrid ‘HTML5’ Cross-Platform Apps)

#### 2.1. Aplicación multiplataforma nativa

Una aplicación multiplataforma nativa es aquella que utiliza otro lenguaje de programación y otro conjunto de funciones llamadas API's diferente al que los creadores de la plataforma tenían diseñado para crear aplicaciones. Por ejemplo, para desarrollar aplicaciones para el sistema operativo Android, se nos provee de un lenguaje base para programar, que en este caso es Java, sin embargo los creadores de la plataforma generalmente hacen posible que se puedan integrar más lenguajes o API's (Application Programming Interface) para que podamos realizar nuestras apps en otro lenguaje, Kotlin, por ejemplo para el caso de Android.

Generalizando la idea anterior, para que podamos tener una aplicación multiplataforma, es decir que esta se pueda exportar a más de una arquitectura, lo que sucede es que un proveedor externo crea una API unificada sobre el SDK (software development kit, herramientas necesarias para compilar, debuggear y ejecutar nuestras aplicaciones) de la plataforma en cuestión para que de esta manera se puedan admitir múltiples sistemas operativos con una única base de código.

Por último, el encargado de hacer el trabajo sucio, es decir, de hacer lo necesario para que con un solo código base podamos ejecutar la aplicación en diferentes plataformas es el IDE (Entorno de desarrollo Integrado). El IDE es básicamente un conjunto de tres herramientas, un editor de texto plano, un debugger o depurador y SDK en donde vienen integradas las API's necesarias para que el IDE compile y ejecute nuestro programa y para nuestro particular, para que pueda exportar nuestra aplicación a distintas plataformas.



## 2.2. Aplicación multiplataforma híbrida

Las plataformas Android, iOS y Windows Phone, principalmente, tienen un poderoso y avanzado navegador integrado en su SDK llamado WebView. Al aprovechar todo el potencial de este componente, los programadores pueden usar tecnologías web HTML5 estándar para diseñar y programar partes de su aplicación. Entonces, al final, la aplicación se compone de al menos un marco nativo y HTML / JavaScript ejecutado en un WebView, por lo que se los llama "híbridos".

Las funciones de la aplicación que requieren la entrada del sensor del dispositivo electrónico, como la geolocalización, la cámara o las funciones de nivel inferior, como acceder al sistema de archivos, generalmente utilizan algún puente JavaScript-to-native proporcionado por el marco de aplicaciones híbridas.

## 3. Herramientas para el desarrollo multiplataforma

Para que usted como desarrollador pueda utilizar herramientas de desarrollo multiplataforma es necesario que conozca qué tecnologías implementan, cuál su lenguaje base y por supuesto, lo más importante, identificar las restricciones o limitaciones de la herramienta, ya que como se mencionara con más detalle en la sección de ventajas y desventajas del desarrollo multiplataforma habrá veces que nuestra aplicación sea tan robusta o especializada que migrarla entre plataformas debe de hacerse completamente desde cero.

### 3.1. Herramientas para aplicación multiplataforma nativa

#### 3.1.1. Microsoft Xamarin

De acuerdo con Business of Apps : 1 Xamarin es una herramienta de desarrollo multiplataforma que permite a los desarrolladores crear aplicaciones nativas de iOS y Android, así como aplicaciones de Windows y Mac, utilizando una sola base de código C# compartido. La compañía también permite a los desarrolladores probar aplicaciones en cientos de dispositivos a través de su servicio Xamarin Cloud, ofrece su propio IDE de Xamarin Studio y ejecuta clases en línea en vivo con su programa Xamarin University. (Consultado 16 abril 2018). Las principales empresas multinacionales que utilizan los servicios de Xamarin son :

Aunque hay alrededor de 15,000 empresas que son clientes de ésta poderosa empresa. Lo más importante de ésta plataforma es que tanto el API para iOS como para Android viene en su 100 %, es decir, que cualquier función que necesitemos propia de android o iOS la podremos implementar. Se puede descargar una versión gratuita del IDE que ofrecen y trabajar con ella de manera muy eficiente.

#### 3.1.2. RubyMotion

RubyMotion es una implementación del lenguaje Ruby para la creación de aplicaciones, trabaja exclusivamente en MacOS y ofrece soporte para las API's de iOS, android y el propio MacOS.

Si eres un linuxero o eres alguien que esté acostumbrado a usar la línea de comandos, RubyMotion es tu herramienta ideal ya que, al igual que C o Java es posible compilar desde línea de comandos es por ello que se puede implementar en cualquier editor de texto plano aunque cuenta con un IDE llamado RubyMine que cuenta con un simulador visual.

Ruby es fácil de aprender y su naturaleza dinámica facilita la escritura de abstracciones en capas y lenguajes específicos de dominio (DSL).



Para hacer desarrollo multiplataforma en Android Studio basta con instalar un plugin al IDE. En Android Studio podremos sacar partido de todo el desarrollo nativo, ya que no estamos cambiando de lenguaje de programación ni de IDE. Tan sólo deberemos tener en cuenta aprovechar este motor para reutilizar la mayor parte del código posible.

Toda la información respecto al desarrollo multiplataforma puede ser encontrado en la documentación oficial de android en <https://developer.android.com/index.html>

Además admite el uso de lenguajes de programación como C++ y Java gracias a la caja de herramientas Android NDK

Multi-OS Engine el complemento independiente que se puede integrar a Android Studio, es una tecnología de vista previa que permite a los desarrolladores utilizar su experiencia en Java para desarrollar aplicaciones móviles nativas para iOS \* y Android \* en máquinas host de desarrollo Windows \* y / o OS X \* sin comprometer el aspecto, la sensación y el rendimiento nativos.

### **3.2. Herramientas para aplicación multiplataforma híbrida**

#### **3.2.1. Adobe PhoneGap**

La herramienta ahora propiedad de Adobe se basa en el proyecto de código abierto Apache Cordova y es completamente gratuito, lo que de alguna manera explica su popularidad. Los desarrolladores de PhoneGap pueden escribir aplicaciones una sola vez en HTML, CSS y JavaScript y desplegarla en diferentes dispositivos móviles sin perder las características de una aplicación nativa. PhoneGap es la herramienta de desarrollador basada en la nube construida sobre el marco, que ofrece desarrollo de aplicaciones móviles basadas en la nube sin la necesidad de SDK, compiladores y hardware.

#### **3.2.2. AppMachine**

Con appMachine se pueden aplicaciones personalizadas con nuestros más de 35 bloques de construcción precodificados y combinados, y con sus características de diseño y selección de piel. Importe sus propios datos de un archivo Excel o conecte sus servicios web a su aplicación.

## **4. Ventajas y desventajas del desarrollo multiplataforma**

El desarrollo multiplataforma nos ofrece la posibilidad de programar un aplicación móvil para las distintas plataformas que existen hoy en día (Android y IOS principalmente) con un solo lenguaje de programación, lo cual nos ahorra tiempo y nos permite ofrecer nuestros servicios a un público más grande. Sin embargo debemos de considerar aspecto muy importantes, tales como el rendimiento del hardware que permite ejecutarse a nuestra aplicación, es decir, si queremos crear una aplicación móvil y de escritorio, puede que para escritorio funcione bien debido a que el hardware es potente, sin embargo al migrarlo a un plataforma menos robusta, quizá puede presentar inconvenientes en cuanto al rendimiento de nuestra aplicación.

Por otra parte, el desarrollo multiplataforma brinda más beneficios durante el período de mantenimiento. Si se encuentra un error en una base de código común, debe solucionarse solo una vez. Esto implica ahorrarse cantidades inimaginables de tiempo y de personal especializado en cada sistema donde se haya implementado la aplicación.

Otra de las cuestiones con las que se tiene que tener cuidado es que HTML5 y CSS requieren de un proceso llamado renderizado el cual usa mucho recursos de CPU que en este caso siempre se traducen en un consumo significativo de la batería de nuestros dispositivos electrónicos.



Por último, las actualizaciones hacia cualquier ente de software son inevitables y el SDK de las plataformas móviles no está exento de ello. El problema es que como se explicó detalladamente en la sección de ¿Qué es una aplicación multiplataforma? lo que se hace para poder tener una aplicación multiplataforma es poner API's que dependen directamente del SDK.

Por lo tanto si quiere actualizar la aplicación construida por medio del cross-platform, para que ésta tenga la última versión de SDK, también se deberá modificar la API, lo cual resulta ser un proceso bastante complicado y lleno de problemas porque hay veces que se cambian las implementaciones de las herramientas de SDK, lo cual implica cambiar la implementación del API y a posteriori, probablemente, la de la aplicación.